1. Consider the sequential circuit shown below. Assume that the flip flops have a setup time of 2 ns, a hold time of 1 ns and a propagation delay between 1 and 3 ns. Also assume that the maximum clock skew is 1 ns and that all the gates have a propagation delay between .5 and 2 ns. What is the shortest clock period for which we can be certain that there are no violations of setup times, assuming no changes at the input X? Is the circuit subject to hold time violations? If so, what would you do to eliminate the hold time violations? If the clock goes high at time 0, during what time period must \( X \) be stable to ensure that there are no violations of setup and hold times? During what time period is it possible for the output to be changing. If the clock could change anytime between \( t = -1 \) ns and \( t = +1 \) ns, how do the last two answers change?

![Sequential Circuit Diagram](image)

2. For the sequential circuit in problem 1, construct a state table and a state transition diagram. Note that this is a Moore model circuit.
Design a serial multiply-by-5 circuit. Your circuit will have a single data input \( D \), a synchronous reset input \( R \) and a single output \( Q \). Whenever \( R \) is high, your circuit should output 0, but when \( R \) drops low, your circuit should treat input \( D \) as a binary value, received with the least significant bit first. Your circuit should output a value which is equal to the input value, multiplied by 5. So if you receive the values 0001101 (where the rightmost bit is received on the first clock tick following reset, then the second bit from the right, etc.) your circuit will output 1000001.

Design your circuit as a Moore model circuit. Carry out each of the following steps:

(a) create a state transition diagram for the circuit,
(b) determine the number of flip flops needed and select a state assignment,
(c) write logic equations for each flip flop input,
(d) write logic equations for the output \( Q \),
(e) draw a schematic for your circuit and
(f) do a timing analysis of your circuit; specifically, determine if there are any hold time violations and if so, explain how to fix them, determine the smallest clock period that it can handle without risk of setup time violations, determine when the input must be stable relative to the rising clock edge and determine the time period following a clock transition when the output can be changing (use the timing parameters from problem 1).

4. Design a 4-bit counter that will count in either binary or BCD based on an input \( T \) (\( T=0 \), count up in binary; \( T=1 \), count up in BCD). Your counter should also have an enable input and a synchronous reset input. The counter counts when enable is high and will reset to zero (0000) when reset is low. First, find the next-state equations for this counter. Second, write VHDL for your counter implementing the equations that you came up with. Third, write behavioral VHDL that implements the same functionality (that is, give the functional model and don’t implement each bit’s equations… use “+” to increment, for example). Show a simulation that demonstrates that reset works, enable works and that you properly count and roll over appropriately depending on the value of \( T \). Be sure to simulate both designs!

Hint: At first blush this looks like a very hard problem; You have seven-variable K-maps to deal with! STOP! THINK! How would you implement reset? What about enable? These are simply AND gates on the input or MUXes. Ignore them. Simply find the next states based on the current state and \( T \)! Once you do, you can implement your design adding if-then statements in your VHDL.